

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Using Visual Cues on DOITree for Visualizing Large Hierarchical Data

Quang Vinh Nguyen¹, Simeon Simoff¹ and Mao Lin Huang²

¹MARCS Institute & School of Computing, Engineering & Maths, University of Western Sydney

²School of Software, Faculty of Engineering & IT, University of Technology, Sydney
{q.nguyen@uws.edu.au, s.simoff@uws.edu.au, Mao.Huang@uts.edu.au}

Abstract

This paper extends a previous work on node link tree visualization and interaction by providing visual clues on hidden structures. We adopt the effectiveness of DOITree, a multi-focal tree layout algorithm, for exploring large hierarchical structures. The advantages of visualization are its most familiar mapping for users, its capability on providing multiple focused nodes, and its dynamic rescaling of substructures to fit the available space. By providing various methods of topological previews of substructure including simple icon view, tree view and treemap view, we provide better understanding the topology of hidden branches.

Keywords--- Tree visualization, hierarchical visualization, exploration, interactive visualization.

1. Introduction

In reality, many interesting collections of information are organized in hierarchical forms, such as the organizational structure of a file system, the structure of a classification system, the organization structure, the taxonomy of objects, such as animals, plants, airplanes, etc. Such hierarchical structures not only play significant roles in their own right, but also provide means for representing the structure of a complex domain in a simplified form [1]. These hierarchical structures are often very large with thousands or even millions of elements and relationships. Interactive visualization of hierarchical structures, with capability for deep exploration at different levels of granularity, is crucial for the analysts in the knowledge discovery process. For instance, in computer forensics, visualization of file systems can assist in identifying the suspected regions for deeper investigation.

Various two-dimensional hierarchical structures (or trees) visualization techniques have been proposed, classified as connection, enclosure and hybrid [2]. Effectiveness of each approach is primarily evaluated by the properties of the data in a particular application domain. The connection approach, such as *Classical*

Hierarchical View [3], *Radial View* [4], *Balloon View* [5], *Rings* [6], *Space-Tree* [7], and *Hyperbolic Browser* [8], uses a node-link diagram that displays the relationships in information explicitly. Enclosure or space-filling approach, such as *Treemaps* [9-11], is considered to be successful method for visualizing large hierarchical data sets with attributed properties. Other techniques also combine both enclosure and connection in their visualization, such *Space-Optimized Tree* [12] and *EncCon Tree* [13]. The traditional algorithms of tree layout were summarized in Di Battista et al [14] and Herman et al [15].

Among the above techniques, classical hierarchical views are universally accepted by various users and applications thank to their easy to understand and interpretable. The algorithms are typically based on the algorithms developed by Reingold & Tilford [3]. The classical hierarchical view uses a modular approach to the positioning of nodes where child nodes are positioned “below” their common ancestor for top-down orientation or positioned on the right-side for left-to-right orientation. Later variations of Reingold & Tilford’s algorithm were presented by Kennedy [16], Herman et al. [17], and Bruggenmann-Klein and Wood [18]. The classical layouts, however, tend to expand over the display area from one dimension which are not suitable to large data sets.

Interaction techniques have been applied to the classical hierarchical views, such as zooming [17], fisheye-view [19] and 1D distortion [20]. Unfortunately, display information is lost when zooming or distortion of views occurs when applying fisheye-view or a distortion technique. Other interactive visualization methods have been proposed to overcome the above limitations. They provide a focus+context view by showing the substructures of interest while shrinking other.

Among those, *SpaceTree* [7] and *DOITrees* [21, 22] are the most popular and acceptable techniques for large tree visualizations. *SpaceTree* combines a zooming environment with conventional tree layouts that can dynamically layout structures to best fit the available screen space. Although *SpaceTree* is an excellent method for exploring large hierarchical data set, the expansion view is only applicable to a single focused branch (or

Uses of visual hints for the hidden branches have proved their effectiveness on exploring large hierarchical structures [7]. *DOITrees* provide simple preview icons for summarizing the topology of the shrunk structures. The visual cues are very limited using unshaded triangles whose sizes are proportional to the weight or number of nodes in the branches. Although some hints of the hidden structures are shown in the view, the visualization did not provide a clear presentation of the hidden structure, such as the number of child nodes nor their properties. This limitation could reduce its effectiveness on interactive visualization and exploration of the large structures.

2. DOI Tree Visualization

When a node is clicked, it becomes a new focus. Smooth slow-in and slow-out animations are then applied to newly visible nodes and to previous visible nodes who becoming hidden respectively. Multiple focuses branches can be expanded by clicking on rooted nodes of the branches while pressing *Ctrl* key. The visualization also supports search and filtering options.

DOITrees technique is very powerful for exploring the large hierarchy. The hierarchical structure is presented clearly on the visualization, with multiple focus branches. Space is reasonably utilized where the visualization maximize the number of levels opened at a time. Unfortunately, the lack of visibility of topology or preview of hidden structures prevents users from gaining understanding on the hidden braches. Without a good topological view of shrunk structures, the effectiveness is less significant. For example in Figure 1, it is impossible to gain any hint on the property of branch rooted at "Transfusion" (last node from the root "Patient Events"), unless the branch is expanded via the interaction.

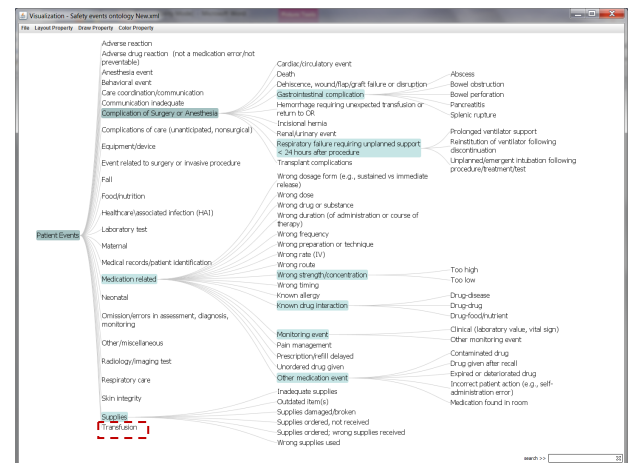


Figure 1. A visualization using DOITrees with no visual cues to preview the topological structure of the hidden branches.

3. Interactive Visualization with Visual Cues

3.1. Simple view.

Sizes of the triangles: sizes of the triangles are proportional to the weights of sub-root nodes or the number of their descendants. The weight of a sub-root node v is calculated recursively from leaves to a vertex using the following formula:

$$w(v) = 1 + C \sum_{i=0}^k w(v_i)$$

Where C is a constant ($0 < C < 1$), and $w(v_i)$ is the weight assigned to the i^{th} child of node v . The weights of all nodes are calculated in prior to the visualization.

To ensure that the visual cues do not oversight the main visualizations, we limit the maximum sizes of triangular shapes.

Shades of the triangles: when a triangle reaches its maximum size, shades are used to represent weights of the large hidden structure. Similarly, the shade darkness is proportional to a node's weight, e.g. the darker of the shade, the more nodes in the structure. The sizes and shading colors of the triangular shapes can be adjusted via an interactive menu.

Figure 2 illustrates a visualization at another navigational state using the same dataset as Figure 1. This view gives a much clearer preview of topological structure of the hierarchy compared to Figure 1. For example, there are a large number of descendants at branches rooted at “Complications of Surgery or Anesthesia”, “Complication of care (unanticipated, nonsurgical)”, “Event related to surgery or invasive procedure” and “Medical related”, indicated by the large black triangles. Although the view gives some insight of the information for the hidden branches, it is not possible to gain further knowledge, such as how many children does the sub-root have? Or what is the property of a child node?

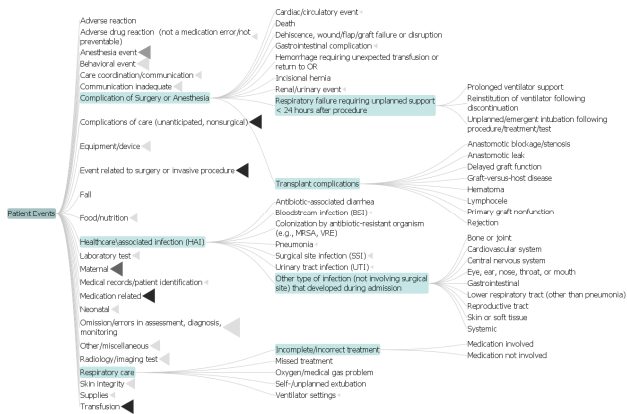


Figure 2. A visualization using DOITrees on the same structure as Figure 2 using the simple view of visual cues to preview the topology of the hidden branches.

3.2. Tree view

The simple *tree view* shows the topology of the hidden branches by providing the classical view for the first level of the branches and the node's property. Each child node in the level is painted with a different color whose shade represents its weight (or the number of descendants of the child node). Similarly to shades of the

triangles, the darker shade, the more weight that the child node has.

Figure 3 illustrates the visualization with the *tree view*. It shows the same data set at the same navigational stage as Figure 2. In this figure, the properties of the first level of the hidden branches are presented quite clearly, such as the number nodes at the next level as well as whether the next level nodes have stronger weight. For example in Figure 2, the branch rooted at “Complication of care (unanticipated, nonsurgical)” (highlighted by a red-dash rectangle) has approximately 20 children, and some of the child nodes also have a large number of descendants.

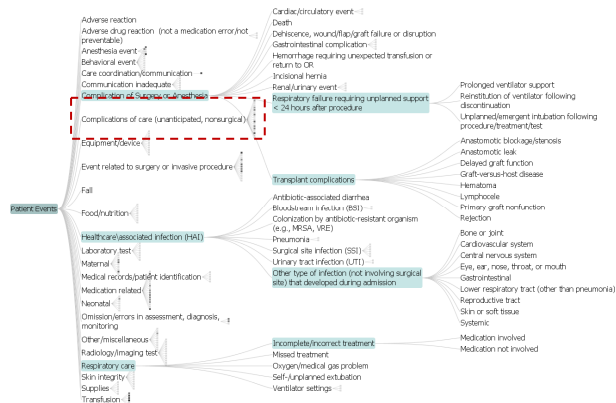


Figure 3. A visualization with the same navigational stage as Figure 2 using the simple tree view of visual cues to preview the topology of the hidden branches.

Although the simple tree view provides better preview of the topological structure, the tree icons might expand significantly when displaying a large number of child nodes. The expansion could create the overlapping with the main view. Figure 4 illustrates the limitation of simple tree view when a node has a large number of children. The expansion of the tree icons create overlapping with the main view, as highlighted by the red-dash rectangles

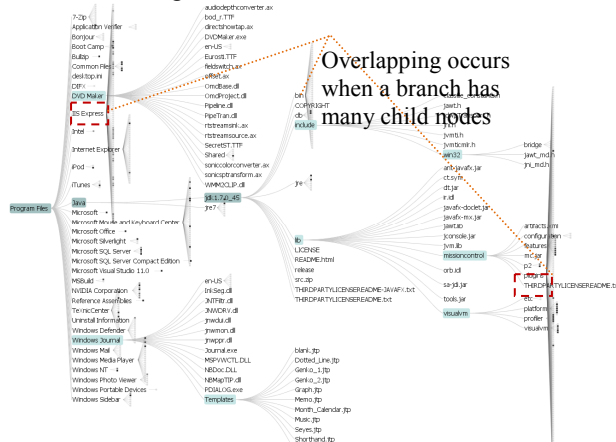


Figure 4. A visualization of a large file system at a navigational stage using the simple tree view of visual cues. The figure shows overlapping might occur when a branch has many children.

3.3. Treemap view

The *treemap view* also shows the topology of the hidden branches by providing the treemap display of the branches and the node's property. We applied a space-filling algorithm [24] on the first level of the hidden branches. This partitioning uses enclosure to represent the substructure, ensuring that all child nodes are located inside sub-root region. Similarly to the simple tree view, sizes and shades represent the weight of child nodes. If a node has more weight, its rectangular icons are larger. Shades are also used to represent the relative weight of child nodes in comparison with other nodes.

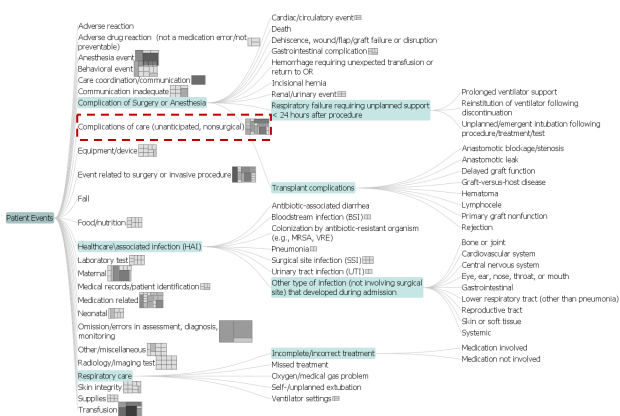


Figure 5. A visualization with the same dataset and navigational stage as Figure 3 using the treemap view of visual cues to preview the topology of the hidden branches.

nonsurgical)” (highlighted by a red dash rectangle). Although the *treemap* view provides better view for comparing node weight due to clear sizes and shades, large rectangular shapes might create overlapping among the treemap icons or it is hard to distinguish nodes in small rectangles when the weight of one or more nodes are very much larger than others (see Figure 6). The size of the visual cues is easily adjusted via a menu.

4. Case studies

We apply our visualization on a various data sets in different applications, such as file systems, medical oncology, product categories, to evaluate the effectiveness of visual cues on the interactive visualizations. The preliminarily informal evaluation confirms that users gain better understanding on the hierarchical structures and perform exploration tasks faster when using visual cues. However, it is still unclear which visual cue, *simple view*, *tree view* or *treemap view*, is most the advantage for previewing the topology of the hidden structures. The choice is dependent on the nature the datasets and the preference of the users. Further and formal usability studies are important to evaluate how and what visual cues can be used effectively on the visualizations.

Figure 7 shows a visualization of a large product categories collected from ebay.com. The dataset has over 20,000 items and 7 levels. Figure 7A shows the visualization at a navigational stage without visual cues. Figures 7B, 7C and 7D present the visualizations at different navigational states using *simple view*, *tree view* and *treemap view* respectively.

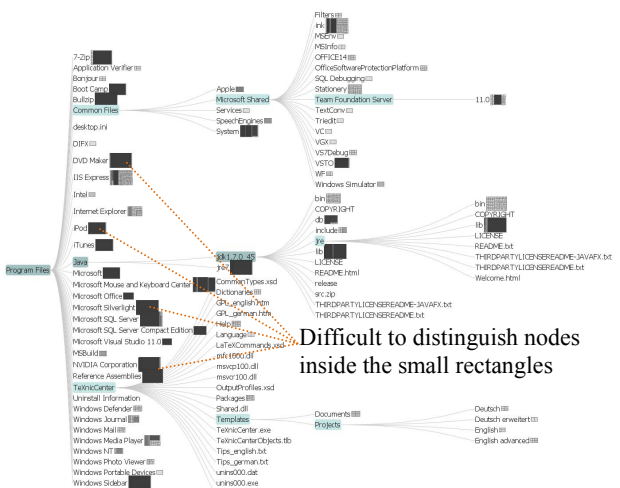
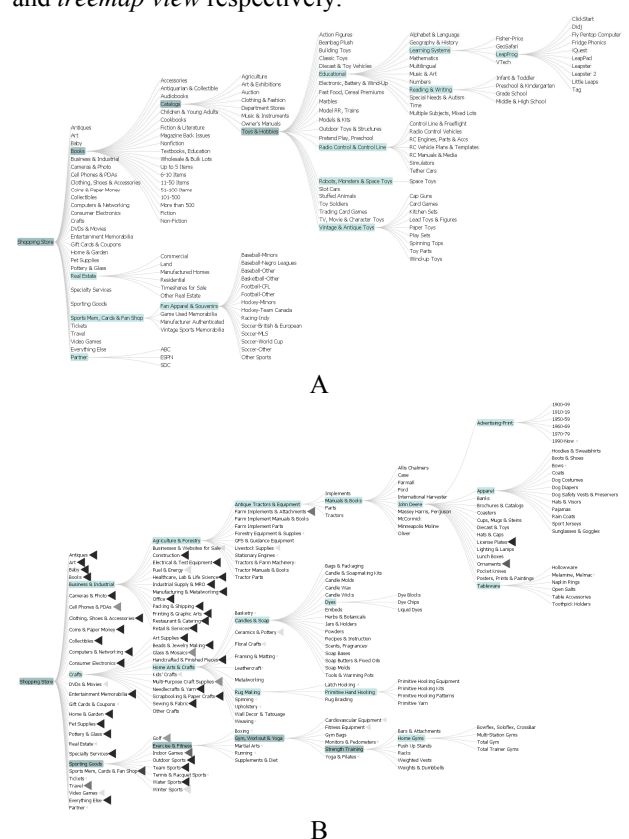
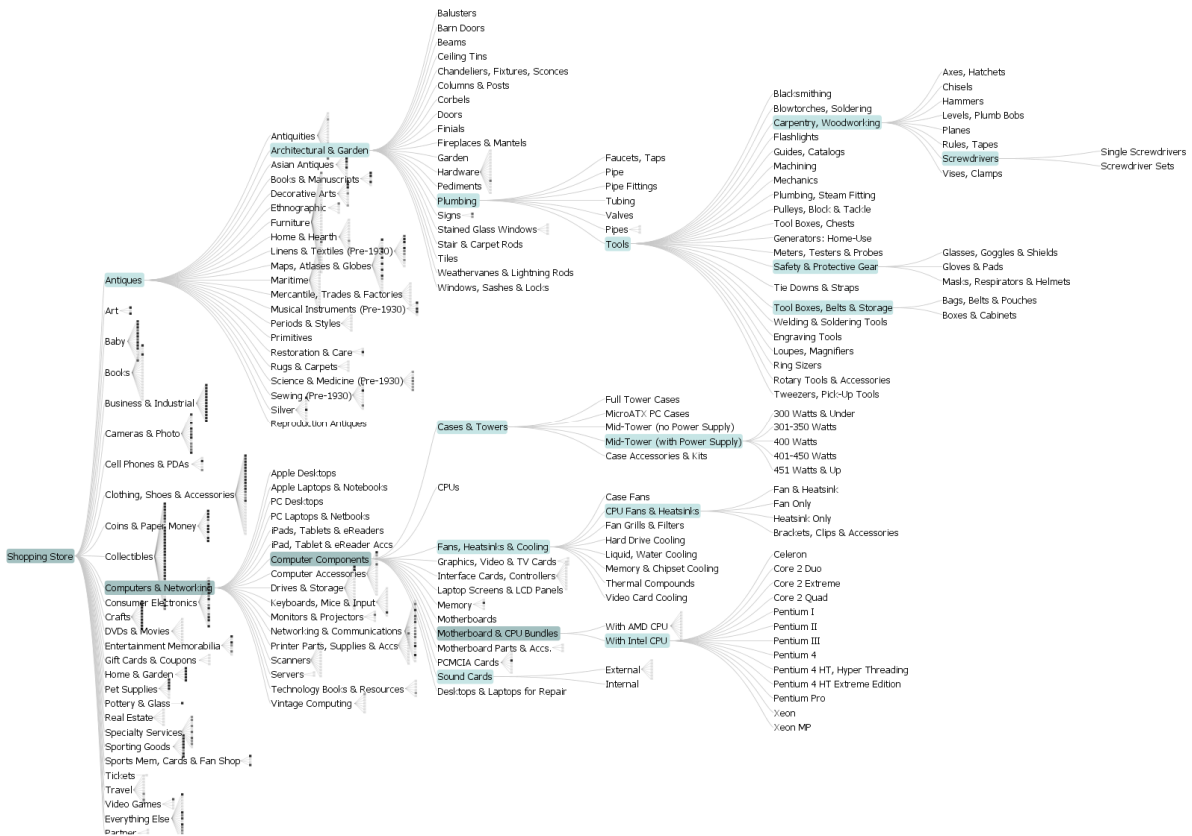


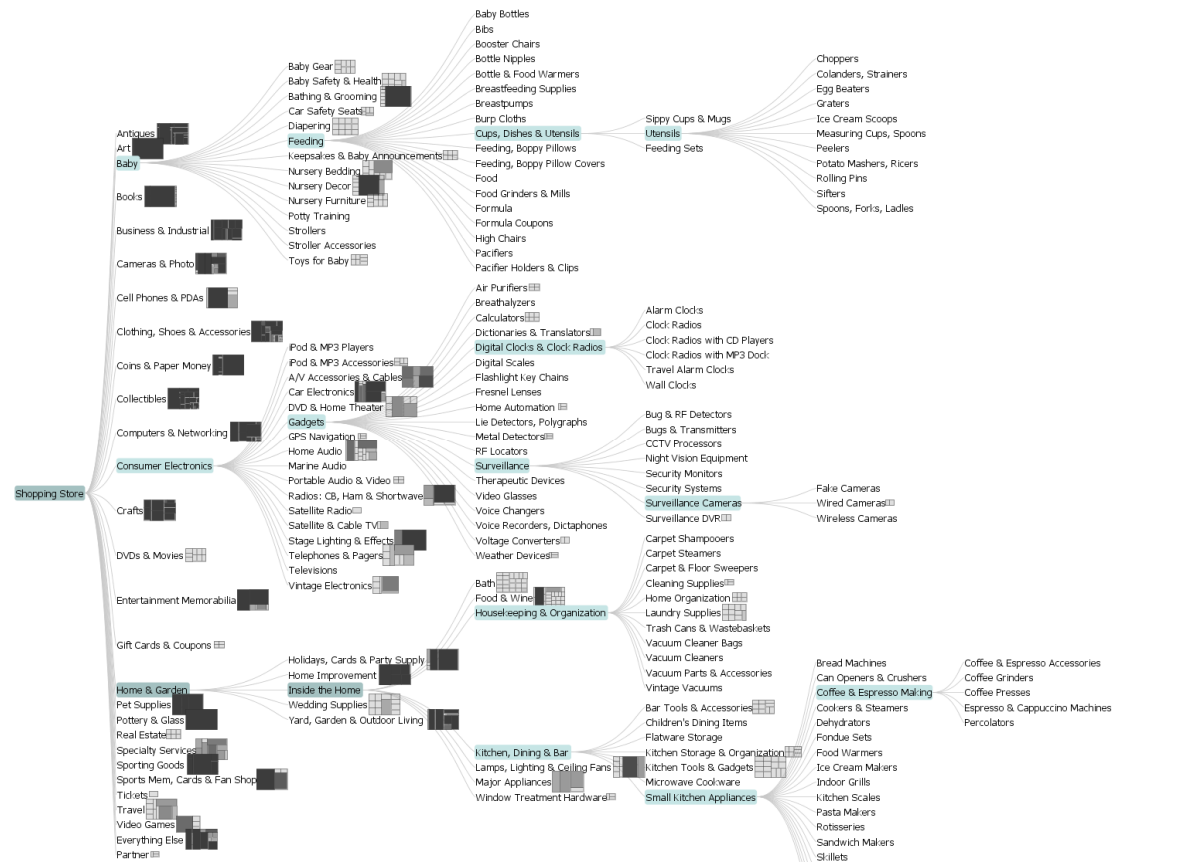
Figure 6. A visualization of a large file system at a navigational stage using treemap view technique. The view shows the problem in identifying nodes inside the small rectangles when the node's weight variation is high.

Our experiments indicate that the *treemap view* presents better attributed properties of the hidden branches. For example, Figure 5 shows clearly the child nodes and their property (i.e. weight), from the branch rooted at “Complication of care (unanticipated,





C



D

Figure 7. A visualization of a large product catalogue (over 20,000 items and 7 levels) at a various state, corresponding to A) no visual cues, B) simple visual cues, C) tree view and D) treemap view.

Conclusions and Future Work

We have presented our visualizations using visual cues on DOITree for exploring Large Hierarchical Data. DOITree is effective and elegant technique who provides multiple focused points using classical node-link views. Unfortunately, the lack of using visual cues previewing of topology of the hidden branches could limit user's understanding on hidden structures, preventing them from gaining better insight during navigation. This paper describes three different methods for showing visual cues on the DOITree visualization, including a simple view of triangular shapes, a simplified node-link tree view and a simplified treemap view.

The preliminary and informal evaluation has proved its effectiveness on using visual cues for exploring large datasets. We will next carry out a formal usability study to evaluate the effectiveness of visual cues on different datasets and applications.

References

- [1] Chen C, *Information Visualization and Virtual Environments*. Springer-Verlag: London, 1999; 223pp.
- [2] Nguyen Q.V., Huang M.L. EncCon: An Approach to Constructing Interactive Visualization of Large Hierarchical Data. *Information Visualization*, 2005, 4(1), pp.1-21.
- [3] Reingold EM and Tilford JS. Tidier Drawing of Trees. *IEEE Transactions on Software Engineering* 1981. 7(2): 223-228.
- [4] Eades P. Drawing Free Trees. *Bulleting of the Institute of Combinatorics and its Applications* 1992, pp. 10-36.
- [5] Melancon G and Herman I. Circular Drawings of Rooted Trees. *Technical Report of the Centre for Mathematics and Computer Sciences* 1998, INS-R9817, Amsterdam, The Netherlands.
- [6] Teoh ST and Ma K-L. RINGS: A Technique for Visualizing Large Hierarchies. *Graph Drawing 2002* (Irvine, California, USA), Springer; pp. 268-275.
- [7] Plaisant C, Grosjean J, and Bederson BB. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *IEEE Symposium on Information Visualization (InfoVis 2002)* 2002 (Boston, MA, USA), IEEE; pp. 57-64.
- [8] Lamping J and Rao R. The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing* 1996; 7(1), pp. 33-55.
- [9] Johnson, B and Shneiderman, B. 1991. Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. *In Proceedings of the IEEE Visualization (Piscataway, NJ, USA)*. IEEE Computer Society, pp. 284-291.
- [10] Bruls, M., Huizing K., and van Wijk, J.J. Squarified Treemaps. *In Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization*, (Vienna, Austria), 2000, Springer, pp.33-42.
- [11] Bederson, B.B., Shneiderman, B. and Wattenberg, M. Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. *ACM Trans. Graphics*. 21(4), 2002, pp. 833-854.
- [12] Q. V. Nguyen, M. L. Huang. A Space-Optimized Tree Visualization. *In Proc. of the IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, IEEE, Boston, Massachussets, October 2002, pp. 85-92.
- [13] Q. V. Nguyen, M. L. Huang. EncCon: An Approach to Constructing Interactive Visualization of Large Hierarchical Data. *Information Visualization*, Palgrave, 4(1), 2005, pp. 1-21.
- [14] Di Battista, G., Eades, P., Tamassia, R. & Tollis, I.G. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New Jersey, 1999.
- [15] Herman, I., Melancon, G. & Marshall, M.S. Graph Visualization in Information Visualization: a Survey. *IEEE Transactions on Visualization and Computer Graphics*. vol. 6, 2000, pp. 24-44.
- [16] Kennedy, A.J. *Functional Pearls. Journal of Functional Programming*. Cambridge University Press, 1996, 6(3), pp. 527-534.
- [17] Herman, I., Delest, M., and Melancon, G. Tree Visualization and Navigation Clues for Information Visualization. *Computer Graphics Forum*, 1998, 17(2), pp. 153-165.
- [18] Bruggemann-Klein, A. and Wood, D. Drawing Trees Nicely with TEX. *Electronic Publishing*, 1988, 2(2), pp. 101-115.
- [19] Furnas, G.W. Generalized Fisheye Views. *ACM SIGCHI '86 Conference on Human Factors in Computing Systems*, Boston, Massachussets, ACM Press, 1986, pp. 15-22.
- [20] Nguyen, Q.V. and Huang, M.L. A Fast Focus+Context Viewing Technique for the Navigation of Classical Hierarchical Layout. *In Proc. of the 7th International Conference on Information Visualisation (IV'03)*, IEEE, London, UK, July 2003, pp. 42-46.
- [21] Card, S.K. and D. Nation. Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface. *In Proc. AVI*. 2002, pp. 231-245.
- [22] Heer, J. and Card, S. K. DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. *In Proc. AVI*. 2004, pp. 421-424.
- [23] Heer, J., Card, S. K., and Landay, J. A. Prefuse: A Toolkit for Interactive Information Visualization. *In Proc. CHI 2005*, Portland, Oregon, USA, ACM, pp. 421-430.
- [24] Liang, J., Nguyen, Q. V., Simoff, S. and Huang, M. L. Visualizing Large Trees with Divide & Conquer Partition. *In Proc. VINCI 2013 – 6th International Symposium on Visual Information Communication and Interaction*, August 2013, Tianjin, China, ACM, 2013, pp. 79-87.